



11

850 101 A

08 735705

SYSTEM AND METHOD FOR CORRELATING PROCESSING DATA
AND IMAGE DATA WITHIN A DIGITAL CAMERA DEVICE

3

4 CROSS-REFERENCE TO RELATED APPLICATIONS

5 This application relates to co-pending U.S. Patent Application
6 Serial No. 08/631,173, entitled "Apparatus And Method For Increasing
7 A Digital Camera Image Capture Rate By Delaying Image Processing,"
8 filed on April 11, 1996, and also relates to co-pending U.S. Patent
9 Application Serial No. 08/666,241, entitled "System And Method For
10 Using A Unified Memory Architecture To Implement A Digital Camera
11 Device," filed on June 20, 1996, which are hereby incorporated by
12 reference.

13

14 BACKGROUND OF THE INVENTION

15 1. Field of the Invention

16 The present invention relates in general to digital camera
17 technology and also relates more particularly to a system and
18 method for correlating processing data and image data within a
19 digital camera device.

1 2. Description of the Background Art

2 An important digital still-camera performance feature is the
3 number of captured images that can be stored in the camera's finite
4 memory. To maximize the image-carrying capacity of digital still-
5 cameras, it is desirable to compress the images prior to storage.
6 Conventional digital cameras typically perform image processing on
7 the captured raw image data and then use a high-quality image
8 compression routine (such as JPEG) to compress the image data.

9 Furthermore, digital cameras may frequently be required to
10 capture and concurrently process multiple successive sets of
11 captured image data. Each captured image, however, has important
12 related information which may be needed during the image
13 processing and compression operations, as well as subsequently.
14 Examples of such processing and compression data might include
15 camera settings (e.g., focus, aperture, and white-balance
16 information), time and date of image capture and image processing
17 parameters.

18 Each captured image potentially has a different set of relevant
19 processing information. Therefore, each captured image within a
20 digital camera may require a separate and unique set of processing

1 data. Furthermore, to permit effective access to these unique sets of
2 processing and compression data, each of the sets of captured image
3 data and the corresponding processing data must be linked together.
4 Efficient access to the processing data at the appropriate time is thus
5 an important feature of modern digital cameras. Therefore, what is
6 needed is an improved system and method for correlating processing
7 data and image data within a digital camera device.

SUMMARY OF THE INVENTION

The present invention is a system and method for correlating processing data and image data within a digital camera device. In the present invention, an imaging device captures an image in response to an image capture request and responsively produces corresponding raw image data which is temporarily stored into a frame buffer. A data cell manager then builds a corresponding data cell containing various types of processing data which the data cell manager links to the captured raw image data. The processing data may include information such as image-capture settings, image size, user tags and image-processing parameters. The data cell is preferably stored in working memory within the camera DRAM.

A first RAM spooler then typically transfers the raw image data into an individual image data file within a RAM disk in the camera DRAM. Next, the data cell manager makes a copy of the data cell in working memory and places the copy into the image data file stored in the RAM disk for recovery purposes. The image data in the frame buffer is then deleted to allow a camera user to capture another image.

1 A first flash spooler next transfers the raw image data file from
2 the RAM disk to a flash memory which preferably is a removable
3 flash disk. An image processor device then accesses, processes and
4 compresses the raw image data using the corresponding processing
5 data stored in the data cell. The image processor device may then
6 directly store the compressed data into a compressed image data file
7 on the RAM disk, or alternately, a second RAM spooler may store the
8 compressed image data into a compressed image data file on the
9 RAM disk. The cell manager then stores selected necessary
10 processing data from the corresponding data cell into the compressed
11 image file. The cell manager also deletes unnecessary processing
12 data from the data cell stored in working memory. A second flash
13 spooler then transfers the compressed image data file from the RAM
14 disk to the flash memory.

15 The data cell thus allows specific camera settings which exist at
16 image capture time to be effectively saved and linked to the
17 corresponding image data, thereby permitting subsequent changes of
18 the camera settings without losing those camera settings previously
19 saved in the data cell. The present invention also allows the camera
20 device to recover from disruptive events such as power failures

1 which threaten to damage the captured image data. Following a
2 disruptive event, the data cell manager may locate the copy of the
3 data cell which is stored in the image data file and then use this
4 copied data cell to rebuild the original data cell stored within
5 working memory in the camera DRAM. Once the original data cell
6 has been reconstructed, the camera may then successfully complete
7 the processing, compression and storage operations for the captured
8 image data.

1 BRIEF DESCRIPTION OF THE DRAWINGS

2 Figure 1 is a block diagram showing a digital camera device
3 according to the present invention;

4
5 Figure 2 is a block diagram showing a preferred embodiment of
6 the FIG. 1 imaging device according to the present invention;

7
8 Figure 3 is a block diagram showing a preferred embodiment of
9 the FIG. 1 computer of the present invention;

10
11 Figure 4 is a block diagram showing a preferred embodiment of
12 a Random Access Memory (RAM) of the Figure 3 computer;

13
14 Figure 5 is a block diagram showing a preferred embodiment of
15 a Read Only Memory (ROM) of the Figure 3 computer;

16
17 Figure 6 is a block diagram showing a preferred embodiment of
18 the FIG. 1 camera device according to the present invention;

1 Figure 7 is a block diagram showing priority levels of preferred
2 processes and corresponding image data paths;

3
4 Figure 8 is a block diagram of the preferred embodiment for an
5 exemplary data cell according to the present invention;

6
7 Figure 9A is the initial portion of a flowchart showing the
8 operation of the present invention using the data cell of Figure 8;

9
10 Figure 9B is the final portion of a flowchart showing the
11 operation of the present invention using the data cell of Figure 8; and

12
13 Figure 10 is a flowchart showing preferred method steps for
14 using the present invention to recover from a disruptive event
15 within a digital camera device.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention comprises a system and method for correlating processing data and image data within a digital camera device and includes an imaging device for capturing image data, a data cell manager for building a data cell containing processing data and for linking the data cell to the captured image data, and a processor device for processing and compressing the captured image data by using the processing data stored within the data cell.

Referring now to Figure 1, a block diagram of a preferred embodiment of a camera 10 is shown. Camera 10 may be used to capture a set of image data representing an object 12. Camera 10 preferably comprises an imaging device 14, an external bus 16 and a computer 18. Imaging device 14 is optically coupled to object 12 and electrically coupled via external bus 16 to computer 18. Once a photographer has focused imaging device 14 on object 12 and, using a capture button or some other means, instructed camera 10 to capture an image of object 12, computer 18 commands imaging device 14 via external bus 16 to capture raw image data representing object 12. The captured raw image data is transferred over external bus 16 to computer 18 which performs various image

1 processing functions on the image data before storing it in its
2 internal memory. External bus 16 also passes various status and
3 control signals between imaging device 14 and computer 18.

4 Referring now to Figure 2, a block diagram of a preferred
5 embodiment of imaging device 14 is shown. Imaging device 14
6 preferably comprises a lens 20 having an iris, a filter 22, an image
7 sensor 24, a timing generator 26, an analog signal processor (ASP)
8 28, an analog-to-digital (A/D) converter 30, a digital signal processor
9 (DSP) 40, and one or more motors 32.

Patent No. 5,496,106,
a 10 ~~U.S. Patent Application Serial No. 08/355,031, entitled "A~~
a 11 ~~System and Method For Generating a Contrast Overlay as a Focus~~
a 12 ~~Assist for an Imaging Device," filed on December 13, 1994 is~~
a 13 ~~incorporated herein by reference and provides a detailed discussion~~
a 14 ~~of the preferred elements of imaging device 14.~~ *In operation* Briefly, imaging
15 device 14 captures an image of object 12 via reflected light
16 impacting image sensor 24 along optical path 34. Image sensor 24
17 responsively generates a set of raw image data representing the
18 captured image 12. The raw image data is then routed through ASP
19 28, A/D converter 30 and DSP 40. DSP 40 has outputs coupled to
20 lines 35, 38 and 42 for controlling ASP 28, motors 32 and timing

1 generator 26. From DSP 40, the raw image data passes over external
2 bus 16 to computer 18.

3 Referring now to Figure 3, a block diagram of a preferred
4 embodiment of computer 18 is shown. Computer 18 comprises a bus
5 interface 52, a processing unit 54, a read-only memory (ROM) 56, an
6 input device 58, a random access memory (RAM) 60, an I/O interface
7 62, a flash memory 64 and a non-volatile memory 68 coupled
8 together via an internal bus 66. In the preferred embodiment,
9 computer 18 is embedded as part of camera 10 using a conventional
10 architecture. However, those skilled in the art will recognize that in
11 an alternate embodiment, computer 18 may be a discrete computer
12 system.

13 Bus interface 52 is preferably a bi-directional first-in, first-out
14 interface for receiving the raw image data and imaging device 14
15 control signals passed between computer 18 and DSP 40. Interface 52
16 has data lines coupled to both external bus 16 and internal bus 66.
17 Processing unit 54 executes programming instructions stored in ROM 56
18 and RAM 60 to perform various operations. ROM 56 stores a set of
19 computer readable program instructions which control how processing
20 unit 54 accesses, transforms and outputs the image data. While ROM 56

1 is employed as a conventional non-volatile memory device for
2 practicing the present invention, those skilled in the art will recognize
3 that in alternate embodiments ROM 56 could be replaced with a
4 functionally equivalent computer useable medium such as a compact
5 disk and drive, a floppy disk and drive, or a flash memory.

6 Input device 58 preferably comprises a series of control
7 buttons which generate signals translated by processing unit 54 into
8 an image capture request, an operating mode selection request, and
9 various control signals for imaging device 14. In an alternate
10 embodiment in which computer 18 is a discrete computer system,
11 input device 58 also includes a keyboard and mouse-type controller.

12 I/O Interface 62 is coupled to internal bus 66 and has an
13 external port connector for coupling computer 18 with a host
14 computer (not shown) for downloading image data stored in RAM 60
15 and/or flash memory 64. At the user's choice or when camera 10 is
16 completely filled with image data, I/O Interface 62 enables the
17 image data to be down-loaded, thus freeing up storage space for
18 future sets of image data.

19 Flash memory 64 serves as an additional image data storage
20 area and is preferably a non-volatile device, readily removable and

1 replaceable by a user. Thus, a user who possesses several flash
2 memories 64 may replace a full flash memory 64 with an empty
3 flash memory 64 to effectively expand the picture taking capacity
4 of camera 10. In the preferred embodiment of the present
5 invention, flash memory 64 is a flash disk. Non-volatile memory 68
6 stores an image counter whose current value becomes an identifier
7 for each new set of image data captured by camera 10. The counter
8 is preferably incremented each time a new image is captured. In the
9 preferred embodiment, non-volatile memory 68 is either an EEPROM
10 or a battery-backed SRAM.

11 Referring now to Figure 4, a block diagram of a preferred
12 embodiment of RAM 60 is shown. RAM 60 is comprised of a frame
13 buffer 70, a working memory 72 and a RAM disk 74. Frame buffer
14 70 preferably comprises a dedicated space of contiguous memory
15 suitable for storing the raw image data generated by image sensor
16 24. In alternate embodiments, frame buffer 70 may be memory
17 space allocated within working memory 72. The function of frame
18 buffer 70 is to store the most recently captured set of raw image
19 data until computer 18 either stores the raw image data in RAM disk
20 74 or transfers it to an image processing unit.

1 RAM disk 74 is a memory area within RAM 60 organized in a
2 "sectored" format similar to that of conventional hard disk drives.
3 The RAM disk 74 function is to store image data. RAM disk 74, in
4 conjunction with flash memory 64, sets the maximum image holding
5 capacity of camera 10. Once both flash memory 64 and RAM disk 74
6 have been filled with compressed image data, the insertion of a new
7 flash memory 64 or down-loading the image data via I/O interface
8 62 will enable camera 10 to continue capturing new images.

9 Working memory 72 is comprised of data cells 76, input
10 queues 78, storage status 80 and temporary buffer 81. Data cells 76
11 are data structures and each data cell 76 is uniquely associated with
12 particular captured image data. A data cell 76 is comprised of a
13 plurality of data cell elements which are further described below in
14 conjunction with FIG. 8. Input queues 78 are data structures
15 comprised of a plurality of data cell "pointers" each corresponding to
16 data cells 76. In the preferred embodiment, input queues operate on
17 a first-in/first-out basis.

18 Storage status 80 is a data structure describing the remaining
19 available memory in both RAM disk 74 and flash memory 64.
20 Storage status 80 contains the following four conditional variables:

1 "RAM Disk Raw File Space," "RAM Disk Compressed File Space," "Flash
2 Memory Raw File Space" and "Flash Memory Compressed File Space."
3 Each of the four conditional variables is set to one of three values:
4 FULL, ALMOST FULL or OK. If the variable is set to "OK," then space
5 is available for that particular file type (i.e., a raw file or a
6 compressed file) on that particular storage resource (i.e., RAM disk
7 74 or flash memory 64). If the variable is set to "ALMOST FULL"
8 then space is not currently available for that particular file type on
9 that particular storage resource, but there will be space in the future.
10 If the variable is set to "FULL" then, absent an increase in available
11 space on storage resources (due, for example, to downloading data or
12 replacing storage units), no space is available for that particular file
13 type on that particular storage resource, nor will space be available
14 in the future. Temporary buffer 81 of working memory 72 is
15 provided for temporarily storing data and/or program code.

16 Referring now to Figure 5, a block diagram of a preferred
17 embodiment of ROM 56 is shown. ROM 56 preferably contains code
18 for processes 82 through 96, including a control application (CA) 82,
19 a RAM spooler 1 (RS1) 84, a flash memory spooler 1 (MS1) 86, image
20 processing/compression (IPC) 88, a RAM spooler 2 (RS2) 90, a flash

1 memory spooler 2 (MS2) 92, a file manager 94, a data cell manager
2 95 and an operating system 96. In alternate embodiments, the
3 Figure 5 processes 82 through 96 may be stored in various computer
4 memory types other than ROM 56.

5 A "spooler" is herein defined as a routine for transferring data
6 from one process or device to a second process or device. RAM
7 spooler 1 (84) transfers raw image data into RAM disk 74, and flash
8 memory spooler 1 (86) transfers raw image data into flash memory
9 64. RAM spooler 2 (90) transfers compressed image data into RAM
10 disk 74 or to I/O interface 62, and flash memory spooler 2 (92)
11 transfers compressed image data into flash memory 64.

12 Control application 82 preferably comprises program
13 instructions for controlling the operation of camera 10 which are
14 executed using processing unit 54. For example, control application
15 82 controls ^{data} cell manger ⁹⁵ ~~85~~ to create and maintain data cells 76.

16 Image processing/compression 88 compresses the raw image data to
17 maximize the image-carrying capacity of camera 10, and also
18 processes the raw image data to permit readily displaying the
19 captured image data on a host computer.

1 In the preferred embodiment, processes 82 through 96 are
2 comprised of a series of software steps implemented on top of a
3 multithreaded operating system and may therefore run in parallel
4 operation. Data cell manager 95 controls and coordinates data cells
5 76 and is further discussed below in conjunction with FIGS. 7
6 through 10.

7 Referring now to Figure 6, a block diagram of the preferred
8 embodiment for camera 10 is shown. In Figure 6, frame buffer 70
9 receives and stores raw image data previously captured by imaging
10 device 14. Frame buffer 70 then provides the raw image data via
11 line 100 to rotate process 95 ~~which is described in detail in co-~~
12 pending U.S. Patent Application Serial No. 08/588,210, entitled
13 "Apparatus and Method for Rotating the Display Orientation of a
14 Captured Image," filed on January 19, 1996, which is hereby
15 ~~incorporated by reference.~~

16 Process 95 rotates the captured image if necessary and then
17 transfers control of the raw image data to RAM spooler 1 (84) using
18 line 102. Alternately, if RAM disk 74 is full, rotate process 95 may
19 transfer control of the raw image data directly to image
20 processing/compression (IPC) 88 using line 118. If RAM spooler 1

1 (84) receives control of the raw image data, it then stores the raw
2 image data into RAM disk 74 using line 104.
3 Flash spooler 1 (86) may then access the raw image data from
4 RAM disk 74 via line 106 and store it into flash memory 64 using
5 line 108. Alternately, if flash memory 64 is full, RAM disk 74 may
6 provide the raw image data directly to IPC 88 using line 114. If
7 flash spooler 1 (86) stores the raw image data into flash memory 64,
8 then IPC 88 typically accesses the stored raw image data using line
9 110 and processes the raw data to responsively obtain compressed
10 image data.

11 IPC 88 may bypass RAM spooler 2 (90) and store the
12 compressed data directly to RAM disk 74 via line 115, or alternately,
13 if RAM disk 74 is temporarily full, IPC 88 may write the compressed
14 data to temporary RAM buffer 81 via line 85. RAM spooler 2 (90)
15 may then access the compressed image data via line 87 and write the
16 accessed data into RAM disk 74 via line 104. RAM spooler 2 (90)
17 may also download the compressed image data to I/O interface 62
18 using line 116. Once the compressed data is in RAM disk 74, flash
19 spooler 2 (92) then accesses the data via line 106 and writes the
20 compressed data into flash memory 64.

1 The present invention may thus process and store a sequence
2 of captured images received from imaging device 14. Although the
3 above example traces the typical data path for a single captured
4 image, the present invention may readily operate on a plurality of
5 captured images progressing through various stages of camera 10.
6 Therefore, multiple sets of image data may exist simultaneously
7 within computer 18. The current processing stage for a specific set
8 of image data is preferably indicated by flags located in the image
9 data's unique data cell 76.

10 Referring now to Figure 7, a block diagram of priority levels for
11 processes 84 through 92 of the preferred embodiment is shown.
12 Background processes 84 through 92 are preferably allotted
13 processing unit 54 time depending on their priority level. This
14 priority level is related to the goal of rapidly emptying frame buffer
15 70 to enable rapid capture of successive sets of image data.

16 Control application 82 transfers raw image data from imaging
17 device 14 to frame buffer 70 and may supersede any of background
18 processes 84 through 92. The background process with the highest
19 priority is RAM spooler 1 (84) which moves raw image data out of
20 frame buffer 70 to RAM disk 74. The second highest priority is flash

1 memory spooler 1 (86) which moves raw image data out of RAM disk
2 74 to flash memory 64. The third highest priority is Image
3 Processing/Compression 88 which accesses raw image data and
4 responsively processes and compresses the image data before storing
5 it as compressed image data into RAM disk 74, or if RAM disk 74 is
6 full, into temporary RAM buffer 81 of working memory 72. The
7 fourth highest priority is RAM spooler 2 (90) which, if necessary,
8 may move compressed image data out of working memory 72 into
9 RAM disk 74. The lowest priority is flash memory spooler 2 (92)
10 which moves the compressed image data out of RAM disk 74 into
11 flash memory 64. Those skilled in the art will recognize that either a
12 greater or a lesser number of priority levels than the preferred five
13 may be used in the present invention. Also, alternate embodiments
14 may establish different criteria for routing the captured image data,
15 depending upon memory resources available and/or the maximum
16 image capture rate desired. File manager process 94 and operating
17 system process 96 are not assigned specific priority levels since they
18 either operate in the background or under interrupt conditions.

19 Processes 82 through 92 preferably each has a respective input
20 queue 78(a) through 78(f) which operates on a first-in/first-out

1 basis. If one of processes 82 through 92 has a data cell 76 pointer in
2 its input queue, then only that process can access and perform
3 operations on the image data associated with that particular data cell
4 76. The data cell pointers are passed between processes 82 through
5 92 in a specific order until the original raw image data has been fully
6 processed, compressed and stored in a memory resource.

7 The priority level scheme introduced above may "block" one or
8 more processes 84 through 92 even though a data cell 76 pointer is
9 in its input queue 78. For example, since moving raw image data out
10 of frame buffer 70 has the highest priority, if a user repeatedly
11 captures images in rapid succession, RAM spooler 1 (84) will
12 continue to operate until RAM disk 74 becomes filled with raw image
13 data. While RAM spooler 1 (84) is operating, all of the other lower
14 priority processes 86 through 92 will be "blocked" (i.e., idled), even
15 though some of the lower priority processes 86 through 92 may still
16 have data cell 76 pointers in their input queues 78. This blocking of
17 lower priority processes applies to all priority levels. For example,
18 operation of flash memory spooler 1 (86) will block image
19 processing/compression 88, RAM spooler 2 (90) and flash memory
20 spooler 2 (92), and operation of image processing/compression 88

1 will block RAM spooler 2 (90) and flash memory spooler 2 (92), and
2 so on, until the image data has been fully processed, compressed and
3 stored in memory. Furthermore, if a lower priority ROM process is
4 currently operating and a higher priority ROM process requires
5 processing unit 54, then the lower priority ROM process is
6 immediately blocked until the higher priority ROM process has
7 completed its operations.

8 Referring now to Figure 8, a block diagram of the preferred
9 embodiment for an exemplary data cell 76 is shown. In the
10 preferred embodiment, data cell manager 95 allocates a structure
11 and then builds a separate data cell 76 to correspond with each set of
12 captured image data. An exemplary data cell 76 typically includes
13 data cell elements 800 through 834, however various additional data
14 cell elements may alternatively be included by data cell manager 95.

15 Version number 800 indicates which version of data cell 76 is
16 presently in use, so that data cell manager 95 can work with more
17 than one version of data cell 76. Verification constant 802 is a
18 known constant value used as a check by data cell manager 95 to
19 verify the validity of the data comprising data cell 76. Image name
20 804 identifies the particular captured image which corresponds to

1 data cell 76. Image name 804 is of the preferred form "IMXXXXXX,"
2 where "XXXXXX" is the image number. In the preferred embodiment,
3 the image number "XXXXXX" is not reset, so when images are down-
4 loaded to a host computer, the down-loaded image names 804 will
5 not conflict with image names 804 of image files previously down-
6 loaded to the host computer. However, in an alternate embodiment
7 the image number "XXXXXX" could be reset each time image data is
8 down-loaded from camera 10. Also the "IM" in the image identifier
9 may be replaced with "IO."

10 Image type 806 specifies the format of the captured image.
11 The image type 806 is typically in the form "YYY," is preferably
12 either CFA or JPG which both refer to sets of compressed image data.
13 Image size 808 includes information about the height and width
14 (number of pixels) of the corresponding captured image. Image
15 processing and compression (IPC) 88 uses image size 808 to correctly
16 process a given set of captured image data. User tags 810 include a
17 variety of tags which may be set by a camera 10 user. For example,
18 a user may set a specific user tag 810 to identify whether a
19 particular captured image is a time-lapse image.

1 Folder name 812 contains the name of the specific folder in
2 which a particular captured image file resides. Image status flags
3 814 contain information about how much image processing has been
4 performed on the corresponding image data (e.g., whether the image
5 data is raw or compressed data, and whether the image data is in
6 frame buffer 70, RAM 60, temporary buffer 81 or flash disk 64).
7 The image status flags 814 store either a "Raw Image Data In Frame
8 Buffer" flag, a "Raw Image Data In RAM Disk" flag, a "Compressed
9 Image Data In RAM Disk" flag, a "Raw Image Data In Flash memory"
10 flag, a "Compressed Image Data In Flash memory" flag, or a
11 "Compressed Image Data In Temporary Buffer" flag. Background
12 processing stage 816 indicates the current stage of the image data
13 within the background spooling processes 84 through 92 (FIG. 7).
14 Time/date stamp 818 contains data identifying the time and
15 date that the image data was captured by camera 10. If selected,
16 delete request 820 causes the corresponding image file to be deleted
17 from camera 10. If selected, stop-processing request 822 causes
18 camera 10 to complete the current process and then to temporarily
19 suspend further processing of the corresponding image data.
20 Watermark data 824 selects a particular watermark image and also

1 specifies where the selected watermark is placed on the captured
2 image.

3 Image processing (IP) parameters 826 contain information
4 which IPC 88 uses during processing of the corresponding image
5 data. For example, IP parameters 826 may include the compression
6 level and color depth for a particular captured image. Image-capture
7 settings 828 may include the various camera 10 settings which
8 existed when the corresponding image data was captured. For
9 example, image-capture settings 828 may include camera 10 focus
10 values, shutter speed, aperture, white-balance settings and exposure
11 values.

12 Image data pointer 830 is a pointer to identify the location of
13 the captured image data which corresponds to data cell 76. Error
14 code 832 stores information to indicate whether processing of the
15 image data was successful for each of the background spooling
16 processes 84 through 92. Miscellaneous 834 contains a variety of
17 "housekeeping" information used by data cell manager 95 to control
18 and coordinate the function of exemplary data cell 76.

19 Referring now to Figure 9A, the initial portion of a flowchart
20 showing the preferred operation of the present invention is shown.

1 Initially, camera 10 captures 910 a selected image and stores the
2 captured image data into frame buffer 70. Data cell manager 95
3 responsively builds 912 a data cell 76 in working memory 72 as
4 described above in conjunction with FIG. 8. Data cell manager 95
5 then adds 913 the data cell 76 to a data cell list identifying data cells
6 76 for all captured images. To subsequently access a given captured
7 image, data cell manager 95 typically identifies the data cell 76 for
8 the given image and then locates the corresponding image data file.
9 Next, data cell manager 95 generates a pointer to the location of data
10 cell 76 in working memory 72, and then passes 914 the generated
11 data cell 76 pointer to RAM spooler 1 (84) by placing the data cell 76
12 pointer in the RAM spooler 1 (84) input queue 78(b).

13 RAM spooler 1 (84) then copies 916 the captured image data
14 from frame buffer 70 to RAM disk 74 to create an image data file.
15 Next, data cell manager 95 makes 918 a copy of the data cell 76
16 located in working memory 72 and places the data cell 76 copy into
17 the newly-created image data file on RAM disk 74. RAM spooler 1
18 (84) then deletes 920 the image data from frame buffer 70.

19 Next, data cell manager passes 922 the generated data cell 76
20 pointer to flash spooler 1 (86) by placing the data cell 76 pointer in

1 the flash spooler 1 (86) input queue 78(c). Flash spooler 1 (86) then
2 copies 924 the image data file from RAM disk 74 to flash disk 64 and
3 deletes 925 the image data file from RAM disk 74. Next, data cell
4 manager 95 passes 926 the data cell 76 pointer to image
5 processing/compression (IPC) 88 by placing the data cell 76 pointer
6 in the IPC 88 input queue 78(d). IPC 88 then accesses 928 the image
7 data stored in flash disk 64 to begin the processing and compressing
8 operations. The FIG. 9A method then proceeds to FIG. 9B.

9 Referring now to Figure 9B, the final portion of a flowchart
10 showing the operation of the present invention is shown. Continuing
11 the process steps of FIG. 9A, data cell manager 95 passes 930 to IPC
12 88 a number of pointers for locating specified data cell 76 elements.
13 In the preferred embodiment, the specified data cell 76 elements
14 typically may include IP parameters 826, image size 808, watermark
15 data 824 and image-capture settings 828. IPC 88 responsively uses
16 these received pointers to locate and access those specified data cell
17 76 elements which are needed to effectively process and compress
18 the captured image data.

19 IPC 88 then advantageously uses this information accessed
20 from data cell 76 to process and compress 932 the captured image

1 data. Next, IPC 88 stores 934 the processed and compressed image
2 data into an image data file in RAM disk 74, if space is available. If
3 no space is currently available in RAM disk 74, IPC 88 temporarily
4 stores the compressed image data into another available memory
5 location, such as temporary buffer 81 in working memory 72. In
6 step 934, data cell manager 95 also stores selected data cell 76
7 elements into the compressed image data file created and stored by
8 IPC 88. Data cell manager 95 thus modifies the copy of data cell 76
9 that was placed into the raw image data file during step 918. In the
10 preferred embodiment, the selected data cell 76 elements which data
11 cell manager 95 incorporates into the compressed image data file
12 typically may include image name 804, image type 806, image size
13 808, user tags 810, folder name 812, time/date stamp 818, IP
14 parameters 826, watermark data 824 and image-capture settings
15 828.

16 Data cell manager 95 then deletes 936 unnecessary elements
17 from data cell 76 in working memory 72 to conserve storage space
18 within DRAM 60. The deleted elements have become unnecessary
19 since IPC 88 has already used them to successfully complete the
20 processing and compression operations and since any other relevant

1 elements have been stored in the compressed image file. In the
2 preferred embodiment, the unnecessary data cell 76 elements
3 deleted from data cell 76 in working memory 72 typically include IP
4 parameters 826, image size 808, watermark data 824 and image-
5 capture settings 828. In alternate embodiments, the deleted data
6 cell 76 elements may further include image status flags 814,
7 background processing stage 816, time/date stamp 818, delete
8 request 820 and stop-processing request 822.

9 Next, IPC 88 deletes 938 the raw image data file from flash
10 disk 64, including the copy of the data cell 76 which data cell
11 manager 95 placed into the raw image data file during step 918.
12 Data cell manager then passes 940 the generated data cell 76 pointer
13 to RAM spooler 2 (90) by placing the data cell 76 pointer in the RAM
14 spooler 2 (90) input queue 78(e). RAM spooler 2 (90) then copies
15 942 the compressed image data file to RAM disk 74 if IPC 88 was
16 unable to store the compressed image data file to RAM disk 74 in
17 step 934 above.

18 Data cell manager then passes 944 the generated data cell 76
19 pointer to flash spooler 2 (92) by placing the data cell 76 pointer in
20 the flash spooler 2 (92) input queue 78(f). Flash spooler 2 (92) then

1 copies 946 the compressed image data file to flash disk 64 and
2 deletes 948 the compressed image data file from flash disk 64.

3 FIGS. 9A and 9B illustrate the preferred operation of the
4 present invention using a single captured image and corresponding
5 data cell 76. The present invention, however, typically operates to
6 capture, process and store a series of captured images. Using multi-
7 tasking and task-priority techniques, the present invention may
8 effectively handle multiple captured images at various processing
9 stages within camera 110. Therefore, the process steps of FIGS. 9A
10 and 9B may advantageously be repeated for each captured image in
11 accordance with the present invention.

12 Referring now to Figure 10, a flowchart showing preferred
13 method steps for using the present invention to recover from a
14 disruptive event within camera 10 is shown. Disruptive events may
15 comprise a variety of occurrences which endanger captured image
16 data within camera 10, including a power failure within camera 10
17 or removal of flash disk 64 while a captured image is being
18 processed.

19 Following a particular disruptive event, a camera 10 user
20 initially remedies the disrupting factor and then applies 1010 power

1 to camera 10. Data cell manager 95 then determines 1012 whether
2 an image data file was stored in RAM disk 74 or flash disk 64 prior
3 to the intervening disruptive event. In the event of a power failure
4 within camera 10, image data files on RAM disk 74 are protected
5 through the use of backup batteries.

6 If no image data files were present in RAM disk 74 or flash
7 disk 64 prior to the intervening disruptive event, then the FIG. 10
8 process ends. However, if image data files were present in RAM disk
9 74 or flash disk 64, then data cell manager 95 determines 1014
10 whether the image data files contained raw or compressed image
11 data. If the image data file contained compressed image data, then
12 data cell manager 95 accesses and uses 1016 that compressed image
13 data file to rebuild the data cell 76 in working memory 72.

14 If, however, the image data file contained raw image data, then
15 data cell manager 95 locates 1020 the copy of the data cell 76 stored
16 within the raw image data file and then uses 1022 that raw image
17 data cell 76 to rebuild the data cell 76 within working memory 72.
18 After data cell manager 95 rebuilds the current data cell 76 within
19 working memory 72, then data cell manager 95 determines 1018
20 whether another image file is present in RAM disk 74 or flash disk

1 64. If another image file is present, then the FIG. 10 process returns
2 to step 1014 to rebuild the data cell 76 which corresponds to the
3 additional image file. After data cell manager 95 rebuilds all data
4 cells 76 within working memory 72, camera 110 may then resume
5 1024 normal background spooling processes 84 through 92 to
6 process and store the captured image data.

7 The present invention has been described above with reference
8 to certain preferred embodiments, however those skilled in the art
9 will recognize that various modifications may be provided.
10 Furthermore, while the present invention has been discussed above
11 as applied to digital cameras, those skilled in the art will also
12 recognize that the current apparatus and method may also be
13 applied to various other devices. These and other variations upon
14 the preferred embodiment are provided for by the present
15 invention, which is limited only by the following claims.